

# DAG-width is PSPACE-complete

Saeed Akhoondian Amiri      Stephan Kreutzer  
Roman Rabinovich

December 11, 2014

Berwanger et al. show in [BDH<sup>+</sup>12] that for every graph  $G$  of size  $n$  and DAG-width  $k$  there is a DAG decomposition of width  $k$  and size  $n^{O(k)}$ . This gives a polynomial time algorithm for determining the DAG-width of a graph for any fixed  $k$ . However, if the DAG-width of the graphs from a class is not bounded, such algorithms become exponential. This raises the question whether we can always find a DAG decomposition of size polynomial in  $n$  as it is the case for tree width and all generalisations of tree width similar to DAG-width.

We show that there is an infinite class of graphs such that every DAG decomposition of optimal width has size super-polynomial in  $n$  and, moreover, there is no polynomial size DAG decomposition which would approximate an optimal decomposition up to an additive constant.

In the second part we use our construction to prove that deciding whether the DAG-width of a given graph is at most a given constant is PSPACE-complete.

## 1 Introduction

In the study of hard algorithmic problems on graphs, methods derived from structural graph theory have proved to be a valuable tool. The rich theory of special classes of graphs developed in this area has been used to identify classes of graphs, such as classes of bounded tree width or clique width, on which many computationally hard problems can be solved efficiently. Most of these classes are defined by some structural property, such as having a tree decomposition of low width, and this structural information can be exploited algorithmically.

Structural parameters such as tree width, clique width, classes of graphs defined by excluded minors etc. studied in this context relate to undirected graphs. However, in various applications in computer science, directed graphs are a more natural model. Given the enormous success width parameters had for problems defined on undirected graphs, it is natural to ask whether they can also be used to analyse the complexity of hard algorithmic problems on digraphs. While in principle it is possible to apply the structure theory for undirected graphs to directed graphs by ignoring the direction of edges, this implies a significant information loss. Hence, for computational problems whose instances are directed graphs, methods based on the structure theory for undirected graphs may be less useful.

Tree width is one of the most successful structural complexity measures. It has several characterisations coming from seemingly unrelated notions, e.g., by eliminations orders or cops and robber games. Tree width is also deeply connected to graph minors and has numerous algorithmic applications. The result of several approaches to generalise tree width to digraphs was a number of structural complexity measures for digraphs. Reed [Ree99] and Johnson, Robertson, Seymour and Thomas [JRST01] introduced the concept of *directed tree width* and showed that the  $k$ -disjoint paths problem and more general linkage problems can be solved in polynomial-time on classes of digraphs of bounded directed tree width. Following this initial proposal, several alternative notions of width measures for sparse classes of digraphs have been presented, for instance *directed path width* (see [Bar06], initially proposed by Robertson, Seymour and Thomas), *D-width* [Saf05], *DAG-width* [BDH<sup>+</sup>12] and *Kelly-width* [HK08].

In this work we concentrate on DAG-width. It distinguishes itself in its particularly simple definition. DAG decompositions have a clear structure and the definition of the cops and robber games characterising DAG-width is a straight forward and natural generalisation of the corresponding game for tree width. However, we show some disadvantages of DAG-width.

A crucial task in designing efficient algorithms on graphs where some width is bounded is to find a decomposition of the given graph of small width. Such decompositions, usually trees or DAGs, are used to solve the problem recursively following the decomposition. For tree width and directed tree width one can decompose the graph in fixed parameter tractable time. For D-width and for Kelly-width such algorithms are not known, but there is always a decomposition of polynomial size, so it can be found non-deterministically. Only the complexity of DAG-width was left as an open problem as it was not known whether every digraph has a decomposition of polynomial size.

Surprisingly, in this paper we show that deciding the DAG-width of a digraph is not only not in NP (under standard complexity theoretical assumptions), it is in fact PSPACE-complete. In terms of the DAG-width game this exhibits the worst case complexity of such games. This result is quite unexpected and especially surprising as such a high complexity was to date only exhibited by a form of graph searching games called *domination games* (see [FKM03, FGT11, KO09]). In these games, each cop not only occupies his current vertex (as in other such games) but a whole neighbourhood of fixed radius, which essentially allows to simulate set quantification making the problem PSPACE-complete. The DAG-width game, however, is to the best of our knowledge the only graph searching game with the usual capturing condition that exhibits such a complexity.

With the same proof technique we also show that there are classes of graphs for which any DAG decomposition of optimal width must contain a super-polynomial number of bags. (If  $\text{NP} \neq \text{PSPACE}$ , this would follow from the previous result, but we show this unconditionally.) Furthermore, we obtain that there cannot be a polynomial time approximation algorithm for DAG-width with only an additive error.

## 2 Preliminaries

We assume familiarity with basic concepts of graph theory and refer to [Die12] for background. All graphs in this paper are finite, directed and simple, i.e. they do not have loops or multiple

edges between the same pair of vertices. Undirected graphs are directed graphs with a symmetric edge relation. If  $G$  is a graph, then  $V(G)$  is its set of vertices and  $E(G)$  is its set of edges. For a set  $X \subseteq V(G)$  we write  $G[X]$  for the subgraph of  $G$  induced by  $X$  and  $G - X$  for  $G[V(G) \setminus X]$ . If  $X$  is a set of vertices, we write  $\text{Reach}_G(X)$  to denote the set of vertices reachable from a vertex in  $X$ . If  $X = \{v\}$ , we write  $\text{Reach}_G(v)$ . A strongly connected component of a digraph  $G$  is a maximal subgraph  $C$  of  $G$  which is strongly connected, i.e. between any pair  $u, v \in V(C)$  there are directed paths from  $u$  to  $v$  and from  $v$  to  $u$ . All components considered in this paper will be strong and hence we simply write *component*.

A *DAG decomposition* of  $G$  is a tuple  $(D, B)$  where  $D$  is a DAG and  $B = \{B_d : d \in V(D)\}$  is a set of bags, i.e. subsets of  $V(G)$ , such that

- (1)  $\bigcup_{d \in V(D)} B_d = V(G)$ ,
- (2) for all  $a, b, c \in D$ , if  $a < b < c$ , then  $B_a \cap B_c \subseteq B_b$ ,
- (3) for every root  $r \in V(D)$ ,  $\text{Reach}_G(B_{\geq r}) = B_{\geq r}$  where  $B_{\geq r} = \bigcup_{r \leq d} B_d$ ,
- (4) for each  $(a, b) \in E(D)$ ,  $\text{Reach}_{G-(B_a \cap B_b)}(B_{\geq b} \setminus B_a) = B_{\geq b} \setminus B_a$ .

The width of  $(D, B)$  is  $\max_{d \in V(D)} |B_d|$  and its size is  $|V(D)|$ . The *DAG-width*  $\text{DAG-w}(G)$  of  $G$  is the minimal width of a DAG decomposition of  $G$ .

DAG-width can be characterised by a cops and robber game. played on a graph  $G$  by a team of cops and a robber. The robber and each cop occupy a vertex of  $G$ . Hence, a current game position can be described by a pair  $(C, v)$ , where  $C$  is the set of vertices occupied by cops and  $v$  is the current robber position. At the beginning the robber chooses an arbitrary vertex  $v$  and the game starts at position  $(\emptyset, v)$ . The game is played in rounds. In each round, from a position  $(C, v)$  the cops first announce their next move, i.e. the set  $C' \subseteq V(G)$  of vertices that they will occupy next. Based on the triple  $(C, C', v)$  the robber chooses his new vertex  $v'$ . This completes a round and the play continues at position  $(C', v')$ . As we will see, from all positions  $(C, C', v)$ , i.e. when the cops move from their current position  $C$  to  $C'$  and the robber is on  $v$ , the robber has exactly the same choice of moves from any vertex in the component of  $G - C$  containing  $v$ . We will therefore describe game positions by a pair  $(C, R)$ , or a triple  $(C, C', R)$ , where  $C, C'$  are as before and  $R$  induces a component of  $G - C$ .

Formally, a game is a tuple  $(V, V_0, E, v_0, \Omega)$  where  $(V, E)$  is a directed graph, in which  $V$  denotes the set of all positions and  $E$  the set of moves,  $V_0 \subseteq V$  is the set of positions in which player 0 has to move,  $v_0 \in V$  is the start position and  $\Omega \in V^\omega$  is the winning condition. A play is a sequence  $v_0, v_1, \dots$  such that for all  $i \geq 0$ ,  $(v_i, v_{i+1}) \in E$ . Player 0 wins a play  $\pi$  if it is finite and ends in a vertex  $v \in V_1 := V \setminus V_0$  without successors (so Player 1 has to move, but cannot do this) or  $\pi \in \Omega$ . A (memoryless) strategy for Player 0 is a partial function  $\sigma : V \rightarrow V$  such that for all  $v \in V$  where  $\sigma$  is defined,  $(v, \sigma(v)) \in E$ . Strategies for Player 1 are defined analogously. A play  $v_0, v_1, \dots$  is consistent with  $\sigma$  if for each  $v_i \in V_0$  that has a successor, we have  $\sigma(v_i) = v_{i+1}$ . We say that  $\sigma$  is winning if every play consistent with  $\sigma$  is winning for Player 0 (and analogously for Player 1). We say that a game position is consistent with  $\sigma$  if there is a play consistent with  $\sigma$  which contains the position.

The *DAG-width game*  $\mathbb{G}(\text{DAG}, G) = (V, V_0, E, v_0, \Omega)$  on a graph  $G$  is defined as follows. The set of positions is  $V = \text{Pos}(G) = \text{Pos}_c \cup \text{Pos}_r$  where

$$\text{Pos}_c = V_0 = \{(C, R) : C \subseteq V(G), R \subseteq V(G) \text{ is a component of } G - C\}$$

are cop positions and

$$\text{Pos}_r = \{(C, C', R) : C, C' \subseteq V(G) \text{ and } R \subseteq V(G) \text{ is a component of } G - C\}$$

are robber positions. The set of moves is  $\text{Moves}(G) = \text{Moves}_c(G) \cup \text{Moves}_r(G)$  where

$$\text{Moves}_c(G) := \{((C, R), (C, C', R)) : (C, R) \in \text{Pos}_c, (C, C', R) \in \text{Pos}_r\}$$

are cop moves and

$$\begin{aligned} \text{Moves}_r(G) := & \{((C, C', R), (C', R')) : (C, C', R) \in \text{Pos}_r, (C', R') \in \text{Pos}_c \\ & \text{and } R' \text{ is a component of } G - C' \text{ with } R' \subseteq \text{Reach}_{G-(C \cap C')}(R)\} \end{aligned}$$

are robber moves. The start position is  $(\emptyset, \emptyset, \emptyset)$  and the winning condition for the cops (i.e. for Player 0) is  $\Omega = \text{Fin} \cap \text{Mon}$  where  $\text{Fin}$  is the set of all finite plays and  $\text{Mon}$  defines the monotonicity condition as  $\text{Mon} := \{(C_0, C_1, R_0), (C_1, R_1), (C_1, C_2, R_1), \dots\} : R_i \subseteq R_{i+1} \text{ for all } i \geq 0$ . In other words, the cops all finite plays in which the robber could never visit a vertex that has already been unavailable for him. All other plays are won by the robber.

A cop is *free* in a position  $(C, R)$  if he is outside of the graph (i.e.  $|C| < k$  in the game with  $k$  cops) or on a vertex  $v \in C$  such that  $v \notin \text{Reach}_{G-(C \setminus \{v\})}(R)$ , i.e. removing this cop does not lead to non-robber-monotonicity.

### 3 Big DAG Decompositions

Let  $s, t: \mathbb{N} \rightarrow \mathbb{N}$  be two monotonically non-increasing functions with  $2 \leq s(n) < n/\log n$  and  $2 \leq t(n)$  for all  $n \geq 5$ . We define a class of graphs  $G_n(s, t)$  of DAG-width  $n+1$  such that every DAG decomposition of width  $n+1$  has super-polynomially many bags in the size of  $G_n(s, t)$  (measured in the number of vertices), which is in  $O(n^2 \cdot t(n)) = O(n^2 \cdot t(n))$ . The parameter  $s$  will be used to determine the difference between the optimal width of a DAG decomposition and the best possible width of a polynomial size decomposition. The parameter  $t \geq 2$  is used for fine-tuning. Our proof works already if  $s(n) = t(n) = 2$  for all  $n$  and the reader is invited to assume these values at first. We shall consider what changes if  $s$  and  $t$  are different later.

For  $n \in \{1, \dots, 4\}$ , the graph  $G_1(s, t)$  is a single vertex without edges. For  $n \geq 5$ , the graph  $G_n(s, t)$  is constructed as follows (see Figure 1). Let  $M(n)$  and  $C_i(n)$  for  $i \in \{0, \dots, t(n)-1\}$  be pairwise disjoint sets of vertices, each of  $n - s(n)$  elements. Let  $D^s(n)$  be a set of  $s(n)$  elements disjoint from all  $C_i(n)$  and  $M(n)$ , and let  $N(n) = M(n) \cup D^s(n)$ . Let  $A^s(n)$  be a set of  $s(n)$  new vertices, and let  $B^t(n) = \{b_0(n), \dots, b_{t(n)-1}(n)\}$  be a set of  $t(n)$  new vertices. The graph  $G_n(s, t)$  has vertices

$$V(G_n(s, t)) = V(G_{n-s(n)-1}(s, t)) \cup A^s(n) \cup B^t(n) \cup \bigcup_{i=0}^{t(n)-1} C_i(n) \cup N(n).$$

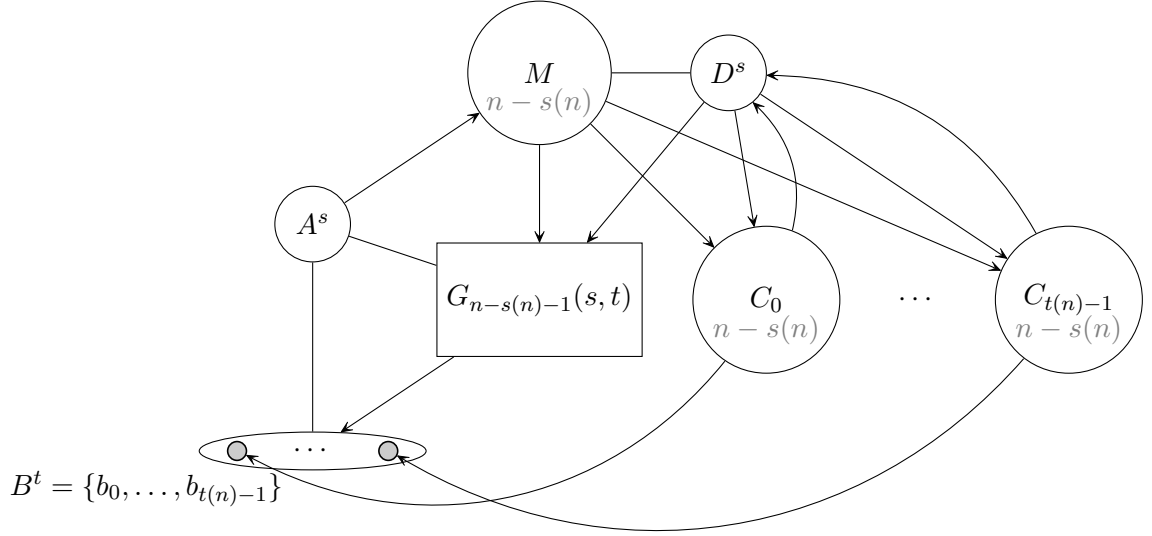


Figure 1: The construction of  $G_n(s, t)$ . Indices  $\cdot(n)$  are omitted.

We say that vertices from  $A^s(n) \cup B^t(n) \cup \bigcup_{i=0}^{t(n)-1} C_i(n) \cup N(n)$  are in level  $n$ . For a set  $X$  let  $\binom{X}{2}$  denote the set  $\{(x, y) \in X^2 : x \neq y\}$ . The edges are defined by

$$\begin{aligned}
 E(G_n(s, t)) = & E(G_{n-s(n)-1}(s, t)) \cup \binom{N(n)}{2} \cup \bigcup_{i=0}^{t(n)-1} \binom{C_i(n)}{2} \cup \binom{A^s(n)}{2} \\
 & \cup \bigcup_{i=0}^{t(n)-1} \left( (N(n) \times C_i(n)) \cup (C_i(n) \times D^s(n)) \cup (C_i(n) \times \{b_i(n)\}) \right) \\
 & \cup (B^t(n) \times A^s(n)) \cup (A^s(n) \times B^t(n)) \cup (A^s(n) \times M(n)) \\
 & \cup (N(n) \times V(G_{n-s(n)-1}(s, t))) \cup (A^s(n) \times V(G_{n-s(n)-1}(s, t))) \\
 & \cup (V(G_{n-s(n)-1}(s, t)) \times A^s(n)) \cup (V(G_{n-s(n)-1}(s, t)) \times B^t(n)).
 \end{aligned}$$

In other words, the first line says that  $G_n(s, t)$  has all edges from  $G_{n-s(n)-1}(s, t)$ , and that  $N(n)$ , all  $C_i(n)$  for  $i \in \{0, \dots, t(n) - 1\}$  and  $A^s(n)$  are cliques of sizes  $n$ ,  $n - s(n)$  and  $s$ , respectively. Note also that  $B^t(n)$  induces an independent set.

For the following lemma the precise definition of  $s$  and  $t$  in  $G_{n-s(n)-1}(s, t)$  is inessential.

**Lemma 1.** *The DAG-width of  $G_n(s, t)$  is  $n + 1$ .*

*Proof.* The cops have the following winning strategy for the DAG-width game on  $G_n(s, t)$ . First they occupy  $N(n)$  and we can assume that the robber chooses some  $C_i(n)$  (for  $i \in \{0, \dots, t(n) - 1\}$ ) because all other strongly connected components of  $G_n(s, t) - N(n)$  have incoming edges from all  $C_i(n)$ . (So the robber can go to every other vertex that is reachable now also later, see Lemma 5.21 in [Rab13].) Then the remaining cop occupies  $b_i(n)$ . If the

robber stays in  $C_i(n)$ , the cops from  $M(n)$  capture him there (recall that  $M(n)$  has the same size  $n - s(n)$  as every  $C_i(n)$ ,  $i \in \{0, \dots, t(n) - 1\}$ ). So we can assume that the robber goes to  $A^s(n) \cup (B^t(n) \setminus \{b_i(n)\})$  (again, there is no reason for him to go directly to  $G_{n-s(n)-1}(s, t)$ ). If the robber remains in  $B^t(n)$ , he is captured in the next move. The cops from  $D^s(n)$  move to  $A^s(n)$  and force the robber to proceed to  $G_{n-s(n)-1}(s, t)$ . From now on, the  $s + 1$  cops in  $A^s(n) \cup \{b_i(n)\}$  stay there until the end of the play and the remaining  $n - s(n)$  cops play in  $G_{n-s(n)-1}(s, t)$  in the same way as on  $G_n(s, t)$  until the robber is captured or expelled to a  $b_j(n)$  for  $j \neq i$ . There he will be captured in one move. Note that  $G_{n-s(n)-1}(s, t)$  has outgoing edges only to  $B^t(n)$  and to  $A^s(n)$ .

A winning robber strategy against  $n$  cops is to stay in  $N(n)$  until all  $n$  cops are there and then to go to  $C_0(n)$ . In that position of the game, no cop can be removed from his vertex due to the monotonicity winning condition.  $\square$

We now prove that the described winning strategy for  $n + 1$  cops (let us call it  $\sigma$ ) is the only possible one up to some irrelevant changes. Then we count the number of positions that are consistent with  $\sigma$  and observe that there are super-polynomially many of them. It will follow that every DAG decomposition of the optimal width has a super-polynomial size.

The first kind of change is to occupy the sets  $M(n)$ ,  $C_i(n)$  ( $i \in \{0, \dots, t(n) - 1\}$ ),  $A^s(n)$  and  $D^s(n)$  in a different order than according to  $\sigma$ , which, obviously, makes no sense, as  $\sigma$  prescribes to occupy either the whole set or none of its vertices and those vertices have ingoing and outgoing edges from and to the same vertices. The second kind of a change is to place cops on and then to remove them from vertices that are already unavailable for the robber. (Note that  $\sigma$  never lets cops stay on such vertices.) This is also useless because of the monotonicity. Both changes can obviously only increase the number of possible positions.

**Lemma 2.** *If there is a DAG decomposition of a graph  $G$  of width  $k$  and of size  $n$ , then  $k$  cops have a winning strategy such that the number of positions consistent with this strategy is at most  $n \cdot |G|$ .*

*Proof.* Consider the strategy (let us call it  $\sigma$ ) obtained from a DAG decomposition of width  $k$  as described in [BDH<sup>+</sup>12, Theorem 16]. In a play consistent with  $\sigma$ , the cops occupy only sets of vertices that correspond to some bag. Thus there are at most  $n$  cop placements that can appear in a play. A position can be described by the cop placement and the robber vertex. There are at most  $|G|$  vertices, so the total number of positions is at most  $n \cdot |G|$ .  $\square$

**Theorem 3.** *Every DAG decomposition of  $G_n(s, t)$  of width  $n + 1$  has super-polynomially many bags.*

*Proof.* We describe the strategy of the robber that enforces the cops to play according to  $\sigma$  up to irrelevant changes. If  $n + 1$  cops play in a different way, they lose.

The robber remains in  $N(n)$  until it is completely occupied by cops. If a cop was placed on a vertex  $v \notin N(n)$  before  $N(n)$  was completely occupied, the cops lose. Indeed, consider the position where all vertices of  $N(n)$  are occupied for the first time. Because  $v$  (whatever it is) has been occupied and because it is still reachable now from  $N(n)$ , the last  $(n + 1)$ -st cop is still on  $v$ , otherwise the monotonicity is violated. The robber goes to some  $C_i(n)$  from which  $v$  is reachable via paths avoiding  $N(n)$  (such a  $C_i(n)$  always exists) and the cops have no legal

move. Thus the first moves of the cops are to occupy  $N(n)$  and the last cop remains outside of the graph.

The robber chooses some  $C_i(n)$  and the cops have no other possible move than to place the last remaining cop on  $b_i(n)$  (otherwise we have the situation discussed in the previous paragraph). The robber goes to  $A^s(n) \cup B^t(n) \setminus \{b_i(n)\}$ . In this position, the cops in  $\{b_i(n)\} \cup M(n)$  cannot be removed and all  $C_i(n)$  are unavailable for the cops. So the cops from  $D^s(n)$  must be used and they can be placed either in  $A^s(n)$  or in  $G_{n-s(n)-1}(s, t)$ , or in  $B^t(n) \setminus \{b_i(n)\}$ . If at least one cop is placed in  $G_{n-s(n)-1}(s, t)$  or in  $B^t(n) \setminus \{b_i(n)\}$ , the robber remains in  $A^s(n)$  until all cops are placed. Then the cops have no legal move and lose. It follows that the cops must occupy the whole  $A^s(n)$  and the robber goes to  $G_{n-s(n)-1}(s, t)$ . From now on, all cops occupying  $A^s(n)$  and  $b_i(n)$  will be reachable from the robber vertex and must stay there. It follows by induction on  $n$  that  $\sigma$  is the unique winning strategy for  $n + 1$  cops up to irrelevant changes.

We now count the number of positions that are consistent with  $\sigma$ . When the robber goes to the last level, the cops are occupying  $A^s(\ell)$  for all levels  $\ell$  that appear as indices of  $G_\ell(s, t)$ . Additionally, for each  $\ell$ , the cops occupy exactly one of  $\{b_0(\ell), \dots, b_{t(\ell)}(\ell)\}$ . (If they occupy more of them, the remaining cops do not suffice to capture the robber due to ??.) Thus when the robber is in the lowest level of the recursion, there are  $t^{\# \ell}$  positions where  $\# \ell$  is the total number of levels in the graph and  $t = \min\{t(\ell) : \ell \text{ is a level in } G_n(s, t)\}$ . As  $s(\ell) < \ell / \log \ell \leq n \log n$ , there are at least  $\log n$  levels, i.e.  $\# \ell \geq \log n$  and the number of bags is at least  $t^{\log n}$ .

The size of  $G_n(s, t)$  is

$$\begin{aligned} |G_n(s, t)| &= |N(n)| + t(n) \cdot |C_i(n)| + |A^s(n)| + |B^t(n)| + |G_{n-s(n)-1}(s, t)| \\ &= n + t(n) \cdot (n - s(n)) + s(n) + t(n) + |G_{n-s(n)-1}(s, t)| = O(n^2 \cdot t(n)), \end{aligned}$$

so  $t^{\log n}$  is super-polynomial in  $|G_n(s, t)|$  for  $\log n \leq t \leq n$ .

Consider a DAG decomposition  $\mathcal{D}$  of  $G_n(s, t)$  of width  $n + 1$ . Towards a contradiction assume that  $\mathcal{D}$  has at most  $|G_n(s, t)|^c$  bags where  $c$  is a constant. Then the winning strategy for  $n + 1$  cops induced by  $\mathcal{D}$  has at most  $|G_n(s, t)|^{c+1}$  positions. We have seen, however, that the number of positions consistent with  $\sigma$  is super-polynomial in  $G_n(s, t)$ .  $\square$

## 4 Consider an Additive Constant Error

In the simplest case we can set  $s(\ell) = t(\ell) = 2$  for all levels. Then we obtain at least  $\lfloor n/2 \rfloor$  levels and the size of an optimal decomposition is at least  $2^{O(\sqrt{|G_n(s, t)|})}$ . However, at the cost of one additional cop we can construct a DAG decomposition with polynomially many bags. We change  $\sigma$  to occupy  $A^s(n)$  with two cops instead of placing one cop on  $b_i(n)$ . We need one extra cop for this, but this is not repeated in each level. Already in the first level, when the robber goes to  $G_{n-s(n)-1}(s, t) = G_{n-3}(s, t)$ , we have cops only on  $A^s(n)$ , but not on  $b_i(n)$ . So one cop is saved for  $G_{n-3}(s, t)$  and we can continue to play in all levels in the same manner.

We can change our choice of  $s$  and  $t$  to make the number of additional cops needed to obtain a polynomial size decomposition unbounded. Let  $s(\ell) = t(\ell) = \lfloor \ell / \log \ell \rfloor$  for all  $\ell$ . Then there

are at least  $\log n$  levels and

$$\begin{aligned}
|G_n(s, t)| &= \sum_{i=0}^{t(n)-1} (|C_i(n)| + |\{b_i(n)\}|) + N(n) + |A^s(n)| + |G_{n-s(n)}(s, t)| \\
&= (n - s(n) + 1) \cdot t(n) + n + s(n) + |G_{n-s(n)-1}(s, t)| \\
&= O\left(\frac{n^2}{\log n}\right) + |G_{n-s(n)-1}(s, t)| = O(n^2).
\end{aligned}$$

It remains to estimate the number of bags in an optimal decomposition. Let  $n_1, n_2, \dots$  be the indices in  $G_{n_i}(s, t)$  appearing in  $G_n(s, t)$ , i.e.  $n_0 = n$ , and for  $i > 0$  we have  $n_i = n_{i-1} - \lfloor n_{i-1} / \log n_{i-1} - 1 \rfloor$ . Then for  $n \geq 5$ ,

$$n_i \geq n - i \cdot n / \log n - i,$$

which is easy to prove by induction on  $i$ . For all  $i \leq \log n / 2$  we have

$$n_i \geq n - \frac{\log n}{2} \cdot \frac{n}{\log n} - \frac{\log n}{2} = \frac{n - \log n}{2}$$

and thus for  $n \geq 5$ ,

$$t(n_i) = \left\lfloor \frac{n_i}{\log n_i} \right\rfloor \geq \frac{(n - \log n)}{2 \log n} \geq \frac{n - n/2}{2 \log n} \geq \frac{n}{4 \log n}.$$

So for  $\lfloor \log n / 2 \rfloor$  many levels  $t(n_i) \geq n / (4 \log n)$  and thus the number of bags in any DAG decomposition is at least  $(\frac{n}{4 \log n})^{\lfloor \log n / 2 \rfloor}$ .

We can define a winning cop strategy with only polynomially many positions with the same trick as before investing  $s(n) - 1$  new cops, i.e. using  $n + s(n)$  cops. Occupy  $N(n)$  and when the robber goes to some  $C_i(n)$ , occupy  $A^s(n)$ . The robber has to go to the lower levels (otherwise he will be captured in  $C_i(n) \cup \{b_i(n)\}$  or in  $B^t(n)$ ) and we do not need cops in  $B^t(n)$ . In the following theorem we show that less than  $n + s(n)$  cops do not have a winning strategy with polynomially many positions. Thus there is no polynomial approximation of an optimal DAG decomposition by an additive constant.

**Theorem 4.** *For all  $G_n(s, t)$  with  $n \geq 25$ , every DAG decomposition of width at most  $n - s(n) - 1$  has size at least  $\left(\frac{\log n}{16}\right)^{\log n / 4}$ .*

*Proof.* We describe a robber strategy against  $n - s(n) - 1$  cops that allows him to enforce at least  $(\frac{n}{4 \log n})^{\lfloor \log n / 2 \rfloor}$  positions (dependent on his choices of  $C_i(\ell)$ ). The robber waits in  $N(n)$  until it is occupied by  $n$  cops and goes to some  $C_i(n)$  for  $i \geq 0$  such that  $b_i(n)$  is not occupied by the cops. As  $|C_i(n)| = n - s(n) \geq s(n)$  (recall that  $n \geq 25$ ) and only  $s(n) - 1$  cops are left, in all  $C_i(n)$  there is a cop free vertex. Similarly, the remaining  $s(n) - 1$  cops cannot occupy all  $b_i(n)$  (there are  $t(n) = s(n)$  many of them), so going to such a  $C_i(n)$  is possible. Now the cops in  $N(n)$  cannot move,  $s(n) - 1$  free cops cannot expel the robber from  $C_i(n)$  and the robber waits in  $C_i(n)$  for  $b_i(n)$  to be occupied. When the cops announce to do this, he runs via  $b_i(n)$



and  $A^s(n)$  (which also has a free vertex) to  $G_{n-s(n)-1}(s, t)$  and plays there in the same way recursively.

If the cops do not occupy all vertices in  $A^s(n)$  when the robber is in the subgraph  $G_{n-s(n)-1}(s, t)$ , they cannot use the cops from  $M(n)$ , so they cannot expel the robber from  $M(n-s(n)-1)$  (i.e.  $M$  in the highest but one level). Indeed,  $|M(n-s(n)-1)| = (n-s(n)-1) - \left\lfloor \frac{n}{\log(n-s(n)-1)} \right\rfloor$  and there are at most  $(n+s(n)-1) - (n-s(n)) - 1 = 2s(n) - 2$  free cops (namely  $n-s(n)$  cops are in  $M(n)$  and 1 cop is on  $b_i(n)$ ), so we only need to choose an appropriately large  $n$ , the least possible being 25. Hence we can assume that the cops occupy all  $A^s(n)$ , i.e.  $s(n) + 1$  cops are tied in level  $n$  and there are at most  $(n+s(n)-1) - s(n) - 1 = n - 2$  cops for  $G_{n-s(n)-1}(s, t)$ .

Let us count the number of possible positions that may appear in a play consistent with the described strategy of the robber. We first count the number of levels  $\ell$  where the cops have more than one cop in  $B^t(\ell)$ . When playing according to  $\sigma$ , which uses  $n + 1$  cops, exactly one cop is in each  $B^t(\ell)$  and now we have  $s(n) + 2$  cops more, so there are at least  $\log n/2$  levels  $i$  with  $t(n_i) \geq n/(4 \log n)$ . Then there are  $\log n/4$  levels  $\ell$  with at most  $4n \log^2 n$  cops in each of them. In order to cover  $B^t(\ell)$  in each such level with  $4n \log^2 n$  cops, we need  $\frac{t(\ell)}{4n/\log^2 n}$  times. As  $t(\ell) \geq n/(4 \log n)$ , we obtain  $\frac{t(\ell)}{4n/\log^2 n} \geq \frac{\log n}{16}$ . Summing up, there are  $\log n/4$  levels where the cops have to choose one among at least  $\log n/16$  placements depending of the robber's choice of the corresponding  $C_i(\ell)$ . Thus the size of any DAG decomposition of width at most  $n + s(n) - 1$  is at least  $(\log n/16)^{\log n/4}$ . Recall that the size of  $G_n(s, t)$  is polynomial in  $n$  for  $t(n) \leq n$ . □

**Corollary 5.** *There is no polynomial size approximation of an optimal DAG decomposition of  $G_n(s, t)$  with an additive constant error.*

## 5 Reducing Tautology in CNF to DAG-width

The construction of  $G_n(s, t)$  shows how the robber can save the history of the play in the current position. We extend it to reduce the TAUTOLOGY problem to DAGW (the problem, given a graph  $G$  and a number  $k$ , is  $\text{DAG-w}(G) \leq k$ ?). TAUTOLOGY is the problem, given a formula of propositional logic, to decide whether it is satisfied by all variable interpretations. In general, TAUTOLOGY is co-NP-hard, but in our version the formula is given in CNF. This is a restriction, as CNF-TAUTOLOGY is in PTIME, but we are going to extend our construction to reduce QBF to DAGW in Section 6 where CNF is the general case and which is more convenient for our purposes. We also restrict the formulae by forbidding a variable to appear twice in a clause.

Let  $\varphi$  be a formula with  $n$  variables and  $m$  clauses. The graphs  $H_\varphi$  are based on the graphs  $G_m(s, t)$  such that

- the number of levels in  $G_m(s, t)$  is one plus the number  $n$  of variables in  $\varphi$ ,
- $s(\ell) = \lfloor \log \ell \rfloor$  and  $t(\ell) = 2$  for all levels  $\ell$ .

It will be convenient to define functions that relate variables and levels. Let the variables of  $\varphi$  be enumerated as  $X_1, X_2, \dots, X_n$ . Let  $\ell: \{1, \dots, n\} \rightarrow \{5, 6, \dots, m\}$  be an injection that maps the index of a variable to a level. Let  $v$  be the function mapping a level to an index of a variable such that  $v(\ell) = i$  if and only if  $\ell(i) = \ell$ .

We replace  $G_4(s, t)$  (which is a single vertex) by the following gadget  $F_\varphi$ . It has a vertex  $v$  and for every clause  $C = L_1 \vee L_2 \vee \dots \vee L_{r(C)}$  an  $r(C)$ -clique  $K_C$  with vertices  $v_1^C, v_2^C, \dots, v_{r(C)}^C$ . The edges go from  $v$  to every vertex of  $K_C$  and back, i.e. we have edges  $(v, v_i^C)$  and  $(v_i^C, v)$  for all clauses  $C$  and all  $i \in \{1, \dots, r(C)\}$ . From outside of  $F_\varphi$ , all vertices that had outgoing edges to the vertex of  $G_4(s, t)$  now have outgoing edges to all vertices of  $F_\varphi$ . So those edges build the set  $A^s(\ell) \times F_\varphi \cup N(\ell) \times F_\varphi$  for every level  $\ell$ . Finally, the edges from  $K_C$  leaving  $F_\varphi$  reflect the clause  $C$ . For all levels  $\ell$  if  $X_j$  with  $j = v(\ell)$  does not appear in  $C$ , then there are edges  $(v_j^C, b_0(\ell))$  and  $(v_j^C, b_1(\ell))$ . If  $X_j = L_i$  for some  $i \in \{1, \dots, r(C)\}$ , then there is an edge  $(v_j^C, b_1(\ell))$ . If  $\neg X_{v(\ell)} = L_i$ , then there is an edge  $(v_i^C, b_0(\ell))$ .

We claim that  $n + 1$  cops capture the robber in  $H_\varphi$  if and only if  $\varphi$  is a tautology. Whether  $\varphi$  is a tautology or not, the cops can play according to the strategy as in the proof of Lemma 1 until the robber component is  $F_\varphi$ . They also have to follow that strategy up to irrelevant changes as was shown in Theorem 3. In that position exactly one cop is free (the one that would capture the robber on  $G_4(s, t)$  if we played on  $G_n(s, t)$ ) and the others occupy all  $A^s(\ell)$  (in all levels) and, in each level  $\ell$ , one of the two vertices  $b_0(\ell)$  and  $b_1(\ell)$ . The free cop is placed on  $v$  in  $F_\varphi$  (there is no other legal move that does not lead to an immediate loss for the cops) and the robber chooses some  $K_C$  for a clause  $C = L_1 \vee L_2 \vee \dots \vee L_r$ . Let  $X_{j_i}$  be the variable in the literal  $L_i$ . Now for all levels  $\ell$  all cops from all  $A^s(\ell)$  cannot be removed and reused as this would violate the monotonicity: there are edges from all vertices of  $K_C$  to all vertices of all  $A^s(\ell)$ . The cops from every level  $\ell$  such that  $X_{v(\ell)}$  does not appear in  $C$  cannot be removed for the same reason. If one cop from some  $B^t(\ell)$  can be reused, the cops win as follows. Let this cop be in level  $\ell$  and assume without loss of generality that  $v(\ell) = j_1$ . The cop is placed on  $v_2^C$ , then the cop from  $B^t(\ell(j_2))$  is placed on  $v_3^C$ . Now the cop from  $B^t(\ell(j_3))$  is placed on  $v_4^C$  and so on.

Then the following holds.

**Lemma 6.** *The cops can win if and only if they can reuse one of  $r$  cops from those  $r$  levels  $\ell$  in which  $X_{v(\ell)}$  appears in  $C$ .*

Assume that  $\varphi$  is true under every valuation. Let  $R \subseteq \{1, \dots, n\}$  be the set of indices  $v(\ell)$  of  $X_{v(\ell)}$  appearing in  $C$ . Let  $\alpha: \{X_i : i \in R\} \rightarrow \{0, 1\}^r$  be the valuation of those  $X_{v(\ell)}$  defined by the choices of the robber during the play as follows. For all levels  $\ell$ , if  $X_{v(\ell)}$  appears in  $C$  and the robber chose the component  $C_i(\ell)$  in level  $\ell$  (for  $i \in \{0, 1\}$ ), then  $\alpha(X_{v(\ell)}) = 1$  if  $i = 0$  and  $\alpha(X_{v(\ell)}) = 0$  if  $i = 1$ . Let  $\beta$  be a valuation of  $X_1, \dots, X_n$  extending  $\alpha$ . As  $\beta \models \varphi$ ,  $\beta \models C$  and thus there is some  $L_j$  in  $C$  with  $\beta \models L_j$ . Let  $\ell$  be such that  $X_{v(\ell)} = L_j$  or  $\neg X_{v(\ell)} = L_j$  (we have  $v(\ell) \in R$ ). If  $X_{v(\ell)} = L_j$ , then  $\beta(X_{v(\ell)}) = \alpha(X_{v(\ell)}) = 1$  and thus the robber chose  $C_0(\ell)$  in level  $\ell$ . Then a cop occupies  $b_0(\ell)$ , but there is no edge from  $K_C$  to  $b_0(\ell)$  (there is an edge to  $b_1(\ell)$ ), so the cop from  $b_1(\ell)$  can be reused and the cops win. If  $\neg X_{v(\ell)} = L_j$ , the situation is symmetric.

If  $\varphi$  is not a tautology, let  $\beta$  be a valuation with  $\beta \not\models \varphi$ . The robber winning strategy is to choose in level  $\ell$  the component  $C_1(\ell)$  if  $\beta(X_{v(\ell)}) = 0$  and the component  $C_0(\ell)$  otherwise.

When the cops occupy  $v$ , the robber chooses the component  $F_C$  corresponding a clause  $C = L_1 \vee \dots \vee L_r$  with  $\beta \not\models C$ . Then  $\beta \not\models L_j$  for all  $j \in \{1, \dots, r\}$ . Thus every cop in all  $B^t(\ell)$  is still reachable from the robber component and the cops lose. By Lemma 6, the cops win.

## 6 DAG-width is PSPACE-complete

We extend our construction again to model choices of the cops in a play that are still recognisable at the end of the play. This leads to a reduction from QBF, which is PSPACE-complete, to DAGW. A *quantified boolean formula*  $\varphi$  is of the form  $\varphi = Q_1 X_1 \dots Q_r X_r \psi(X_1, \dots, X_r)$  where  $Q_i$  is either  $\forall$  or  $\exists$  and  $\psi$  is a propositional formula in CNF with variables from  $\mathcal{X} = \{X_1, \dots, X_r\}$ .

The semantics of  $\varphi$  can be defined by means of a two-player game with perfect information, which is convenient for our reduction. It is the model-checking game  $\text{MCgame}(\varphi)$  for  $\varphi$  on the fixed structure  $(\{0, 1\}, \emptyset)$  with no relations. The players are called  $\forall$  (the universal player) and  $\exists$  (the existential player). A play is played as follows. First, the quantifying prefix of the formula is read from left to right and, for  $i = 1, 2, \dots, r$ , player  $Q_i \in \{\forall, \exists\}$  chooses a value  $\beta(X_i) \in \{0, 1\}$  for  $X_i$ . In other words, we have positions  $P_j$  of the form

$$P_j = Q_j X_j, \dots, Q_r X_r \psi(X_1/\beta(X_1), \dots, X_{j-1}/\beta(X_{j-1}), X_j, \dots, X_r)$$

where  $X_\ell/\beta(X_\ell)$  means that we replace all occurrences of  $X_\ell$  in  $\psi$  by  $\beta(X_\ell)$ . If  $Q_j = \forall$ , then  $P_j$  is a position of the universal player, otherwise  $P_j$  belongs to the existential player. A next position depends on the choice of  $\beta(X_{j+1}) \in \{0, 1\}$  and has the form

$$Q_{j+1} X_{j+1}, \dots, Q_r X_r \psi(X_1/\beta(X_1), \dots, X_j/\beta(X_j), X_{j+1}, \dots, X_r).$$

The remaining positions of the game are of the form  $(\vartheta, \beta)$  where  $\vartheta$  is a subformula of  $\psi$  and  $\beta$  is the valuation of the variables as chosen in first part of the play. The second part starts in position  $(\psi, \beta)$ . If  $\vartheta = \vartheta_1 \vee \vartheta_2$ , the existential player moves to  $(\vartheta_1, \beta)$  or to  $(\vartheta_2, \beta)$  and if  $\vartheta = \vartheta_1 \wedge \vartheta_2$ , then the universal player moves to  $(\vartheta_1, \beta)$  or to  $(\vartheta_2, \beta)$ . In positions  $(X_i, \beta)$ , the existential player wins if  $\beta(X_i) = 1$  and loses if  $\beta(X_i) = 0$ . In positions  $(\neg X_i, \beta)$ , the universal player wins if  $\beta(X_i) = 1$  and loses if  $\beta(X_i) = 0$ . The formula is true if and only if the existential player has a winning strategy in the game.

It is very well known that deciding whether a given quantified formula is true is PSPACE-complete.

The rest of the section is devoted to proof of the following theorem.

**Theorem 7.** *DAGW is PSPACE-complete.*

The easier part is to show that DAGW is in PSPACE. It suffices to prove that any play in the cops and robber game has polynomial length. Then deciding the winner of the game is in APTIME (alternating PTIME) and thus in PSPACE. If  $k$  cops have a winning strategy on a graph, they also have a winning strategy that always prescribes to place cops in a way that the space available for the robber shrinks. We consider a version of the game where the cops have to play in this manner. Then they win if and only if they win in at most  $2n$  moves where  $n$  is the number of vertices of the graph. Thus any play lasts at most  $2n$  steps.

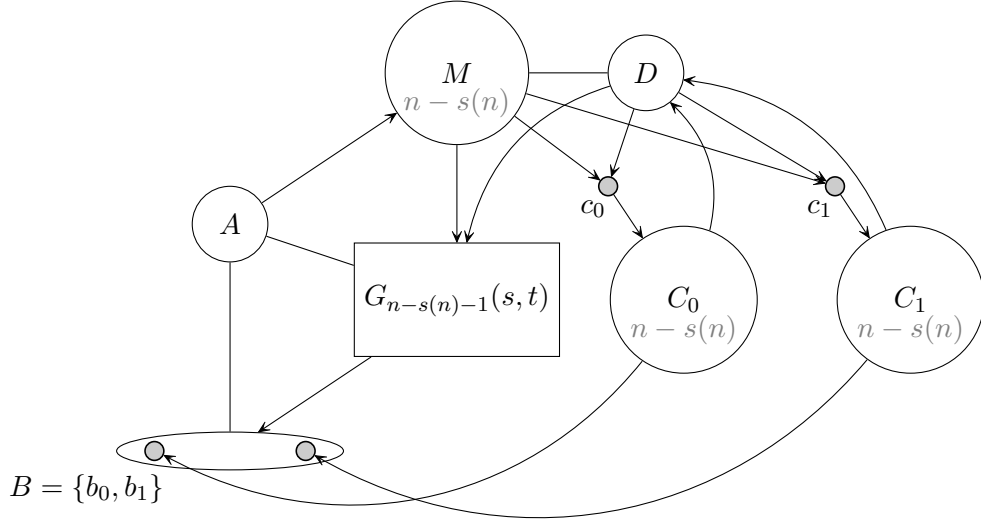


Figure 2: The construction of  $G_n(s, t)$ . Indices  $\cdot(n)$  are omitted.

For the hardness we reduce QBF to DAGW. Let

$$\varphi = Q_1 X_1 Q_2 X_2 \dots Q_r X_r \psi(X_1, \dots, X_r)$$

be a quantified boolean formula. Our construction of the graph  $S_\varphi$  extends the construction from Section 5. For every universal quantifier we add a new level as described in Section 5. For each existential quantifier we add a level that is depicted in Figure 2.

If  $\varphi$  has no variables, then if  $\varphi$  is true,  $S_\varphi$  is a single vertex, and if  $\varphi$  is false,  $S_\varphi$  is a 2-clique. (So one cop wins if and only if  $\varphi$  is true.) Otherwise we start the construction of  $S_\varphi$  with  $F_\psi$  and for  $j = r, r-1, \dots, 1$  we construct graphs  $S_\varphi^j$  such that  $S_\varphi^1 = S_\varphi$ . Assume that  $S_\varphi^{j+1}$  is already constructed, then  $S_\varphi^j$  is the following graph. There are two cases. If  $Q_j = \exists$ , then the vertex set is

$$V(S_\varphi^j) = V_\exists(j) = V(S_\varphi^{j+1}) \cup A(j) \cup B(j) \cup C_0(j) \cup C_1(j) \cup N(n) \cup \{c_0(j), c_1(j)\}.$$

Hereby  $N(j) = M(j) \cup D(j)$  and  $B(j)$  are as  $N(j)$ ,  $M(j)$ ,  $D^s(j)$  and  $B^t(j)$  in  $G_n(s, t)$ , i.e.

$$|B(j)| = |D(j)| = 2, |C_i(1)| = |M(1)| = 4, |C_i(k+1)| = |M(k+1)| = |M(k)| + 3$$

for all  $k \in \{2, \dots, j\}$  and  $i \in \{0, 1\}$ . Furthermore,  $B(j) = \{b_0(j), b_1(j)\}$ .

The set of edges is

$$\begin{aligned}
E(S_\varphi^j) = & E(S_\varphi^{j+1}) \cup \binom{N(n)}{2} \cup \bigcup_{i=0}^1 \binom{C_i(n)}{2} \cup \binom{A(n)}{2} \\
& \cup \bigcup_{i=0}^1 \left( (N(n) \times \{c_i(n)\}) \cup (\{c_i(n)\} \times C_i(n)) \right. \\
& \quad \left. \cup (C_i(n) \times D(n)) \cup (C_i(n) \times \{b_i(n)\}) \right) \\
& \cup (B(n) \times A(n)) \cup (A(n) \times B(n)) \cup (A(n) \times M(n)) \\
& \cup (N(n) \times V(S_\varphi^{j+1})) \cup (A(n) \times V(S_\varphi^{j+1})) \\
& \cup (V(S_\varphi^{j+1}) \times A(n)) \cup E(j).
\end{aligned}$$

Hereby  $E(j)$ , the edges connecting  $F_\psi$  to the new level are defined as follows. Let  $K_C = \{v_1^C, \dots, v_{r(C)}^C\}$  be a clique in  $F_\psi$  corresponding to a clause  $C = L_1 \vee \dots, L_r$ . If  $X_j = L_i$ , then  $(v_i^C, b_1(j)) \in E(j)$ . If  $\neg X_j = L_i$ , then  $(v_i^C, b_0(j)) \in E(j)$ . Otherwise (i.e. if  $X_j$  does not appear in  $C$ )  $\{(v_i^C, b_0(j)), (v_i^C, b_1(j))\} \subseteq E(j)$ .

In the second case  $Q_j = \forall$ . Then  $V(S_\varphi(j)) = V_\forall(j) = V_\exists(j) \setminus \{c_0(j), c_1(j)\}$  and the edges are defined as in  $H_\varphi$ . The difference to a universal level is hence that the edges from  $N(n)$  to  $C_i(n)$  are now subdivided by  $c_i(\ell)$  for  $i \in \{0, 1\}$ .

We are going to show that  $r + 1$  cops win on  $S_\varphi$  if and only if the existential player wins  $\text{MCgame}(\varphi)$ . For that we need some lemmata.

Consider a position  $P$  in the game on  $S_\varphi$  in that there is a level  $\ell$  such that the robber is in  $M(\ell)$ . Let  $b = (b_1, \dots, b_r)$  be a tuple of bits and let  $O^b(\ell) = \bigcup_{\ell' > \ell} A(\ell') \cup \{b_i(\ell') : b_i = 1\}$ . Assume that the cops occupy  $O^b(\ell)$ . Then the robber is blocked in the levels less or equal to  $\ell$  (except that he can go to some  $b_i(\ell')$ , but he would be captured there immediately). Assume that there are  $|N(\ell)| + 1$  free cops in that position.

**Lemma 8.** *For each  $i \in \{0, 1\}$ , the cops have a strategy that allows them either to capture the robber or to expel him from level  $\ell$  such that the cops occupy precisely  $O^b(\ell) \cup A(\ell) \cup \{b_i(\ell)\}$  for an  $i \in \{0, 1\}$ .*

*Proof.* The strategy is as follows. One cop is placed on  $c_{1-i}(\ell)$ . This creates two components of  $S_\varphi^\ell$ : the one induced by  $C_0(\ell)$  and the one induced by  $N(\ell), A(\ell), B(\ell), C_i(\ell)$  and  $S_\varphi^{\ell-3}$ . If the robber is in  $C_i(\ell)$ , the remaining free cops expel him from there and the robber is in the other component. Then the cops occupy  $N(\ell)$  and then the cop from  $c_{1-i}(\ell)$  occupies  $b_i(\ell)$ . If the robber is in  $C_i(\ell)$  and stays there, he is captured there by the cops from  $N(\ell)$ , so assume that the robber goes either to the component induced by  $b_{1-i}(\ell)$  or to the component induced by  $A(\ell)$  and  $S_\varphi^{\ell-3}$ . In any case the cops leave  $D(\ell)$  and occupy  $A(\ell)$ . If the robber remains in  $b_{1-i}(\ell)$ , he is captured by the cop from  $b_i(\ell)$ , so assume that he goes to  $S_\varphi^{\ell-3}$ . We obtain the required position.  $\square$

**Lemma 9.** *The robber has a strategy that permits him either to win or to reach a position from position  $P$  where the robber is in  $S_\varphi^{\ell-3}$  and the cops occupy  $O^b(\ell)$ ,  $A(\ell)$  and at least one of  $b_i(\ell)$ . Furthermore, this robber strategy is winning if there are only  $|M(\ell)|$  free cops.*

*Proof.* The robber stays in  $N(\ell)$  until it is completely occupied by the cops. In that position, one of  $c_i(\ell)$  is not occupied by cops, and the robber runs to a  $C_i(\ell)$  and plays as in the proof of Theorem 3. Note that the cops from  $A(\ell')$  and from  $B(\ell')$  for all  $\ell' > \ell$  cannot be removed.  $\square$

**Lemma 10.** *There is a winning strategy for  $r + 1$  cops on  $S_\varphi$  if and only if  $\varphi$  is true.*

*Proof.* Assume that  $r$  cops have a winning strategy  $\sigma$  on  $S_\varphi$ . In  $\text{MCgame}(\varphi)$ , the existential player simulates the cops and robber game on  $S_\varphi$  by translating the moves of the existential player into robber moves and translating cop moves (according to  $\sigma$ ) into his choices of the existentially quantified variables. Assume that we reached a position  $P$  in the cops and robber game and a position  $P_i$  (for  $i \geq 1$ ) in the  $\text{MCgame}(\varphi)$  such that the following invariant (INV) holds.

- the values  $b_j = \beta(X_j)$  for the variables  $X_j$  where  $j \in \{1, \dots, i-1\}$  are already chosen,
- in the cops and robber game, the robber is in level  $\ell = \ell(i)$  and
- the cops occupy  $O^b(\ell)$  for the tuple  $b = (b_1, \dots, b_{i-1})$  where  $O^b(\ell)$  is defined as above.

Then there are exactly  $\ell+1$  free cops. If  $Q_i = \forall$ , the universal player chooses a value  $b_i = \beta(X_i)$  for  $X_i$ . Then the existential player simulates the cops and robber game playing for the cops according to  $\sigma$  from position  $P$  and for the robber as in Lemma 9 such that the robber is expelled from level  $\ell$  and  $A(\ell)$  and  $b_i(\ell)$  are occupied by cops. The number of free cops suffices for that. It is straightforward to check that the above invariant holds for  $i+1$  and for the position of the cops and robber game where the robber is blocked in the next level, i.e. in level  $\ell-3$ .

If  $Q_i = \exists$ , the existential player simulates the cops and robber game from  $P$  until the robber is expelled from level  $\ell$  according to Lemma 8. Hereby, the cops play according to  $\sigma$  and the robber plays arbitrarily, but such that he is not captured in level  $\ell$ . For example, the robber goes directly to  $S_\varphi^{\ell-3}$ . Again, there are enough free cops for the simulation. Then exactly one of  $b_0(\ell)$  and  $b_1(\ell)$  is occupied by a cop. If it is  $b_0(\ell)$ , the existential player sets  $\beta(X_i) = 0$ , otherwise  $\beta(X_i) = 1$ . Again, the invariant holds.

When all variables have their values, the universal player chooses a clause  $C$  and the universal player simulates in the cops and robber game the move of the free cop to vertex  $v$  in  $F_\psi$  and the move of the robber to  $K_C$ . As the cops have played according to  $\sigma$  and  $\sigma$  is a winning strategy, there is a free cop. The cops completely occupy all  $A(\ell)$  and for all  $\ell$  exactly one of  $b_0(\ell)$  and  $b_1(\ell)$ . Every vertex in all  $A(\ell)$  is still reachable from the robber vertex, so the free cop is in some  $b_i(\ell)$ . As the cop is free, there is no edge from any  $v_j^C$  to  $b_i(\ell)$ , i.e. there is the edge from some  $v_j^C$  to  $b_{1-i}(\ell)$ . By construction of  $S_\varphi$ ,  $X_j$  appears in  $C$  and if  $i = 0$ , then  $X_j$  is negativ and if  $i = 1$ , then  $X_j$  is positiv in  $C$ . In the first case  $b_i = 0$  and  $\beta(X_{v(\ell)}) = 0$ , so  $C$  is satisfied and the existential player wins by choosing  $\neg X_j$ . The second case is symmetric.

For the other direction assume that the existential player has a winning strategy. We show that  $r$  cops have a winning strategy. The cops simulate the game  $\text{MCgame}(\varphi)$  while playing on  $S_\varphi$  by translating he moves of the robber to choices of the universal player and the choices of the existential player to their moves. Assume as before that (INV) holds for some  $i \geq 1$ .

There are two cases: level  $\ell$  is either existential or universal. In any case, there are  $\ell+1$  free cops and if the level is universal,  $\ell$  of them occupy  $N(\ell)$ . The robber escapes to some  $C_i(\ell)$  for an

$i \in \{0, 1\}$  or goes to the component consisting of  $A(\ell)$ ,  $B(\ell)$  and  $S_\varphi^{\ell-3}$ . If the robber is in  $C_i(\ell)$  the last remaining cop is placed on  $b_i(\ell)$  and the robber proceeds to  $\{b_{1-i}(\ell)\} \cup A(\ell) \cup S_\varphi^{\ell-3}$ , otherwise the cop is placed on  $b_i(\ell) = b_0(\ell)$  and the robber is in  $\{b_1(\ell)\} \cup A(\ell) \cup S_\varphi^{\ell-3}$ . Now the cops from  $D(\ell)$  occupy  $A(\ell)$  and the robber goes to  $S_\varphi^{\ell-3}$  (if he goes to the free vertex of  $B(\ell)$ , he loses immediately). Finally, the cops simulate the choice of the universal player:  $\beta(X_{v(\ell)}) = i$ . It is easy to see that the invariant holds.

If the level is existential, the cops look up what value the strategy for the existential player in  $\text{MCgame}(\varphi)$  prescribes to choose for  $X_{v(\ell)}$ :  $\beta(X_{v(\ell)}) = b_i \in \{0, 1\}$ . Then according to Lemma 8 the cops can play such that the invariant holds again.

When the play arrives  $F_\psi$ , there is one free cop that is placed on  $v$  and the robber goes to some  $K_C$  for a clause  $C$ . As the existential player has a winning strategy, there is some literal  $L_j$  in  $C$  that is satisfied by  $\beta$ . Without loss of generality,  $L_j = X_i$  and  $\beta(X_i) = 1$ . Then  $b_1(\ell(i))$  is occupied by a cop, but there is no edge from  $v_i^C$  to  $b_1(\ell(i))$ . Recall that in  $C$  every variable appears at most once, so there is no edge from  $C$  to  $b_1(\ell(i))$  and the cop from  $b_1(\ell(i))$  is free.  $\square$

## 7 Conclusion

We showed that DAG-width cannot be computed efficiently in the classical sense (assuming  $\text{Ptime} \neq \text{Pspace}$ ). It would be interesting to find (fixed-parameter tractable) algorithms that compute reasonable approximations of an optimal DAG decomposition. Another approach to DAG-width would be to show that DAG-width and Kelly-width are bounded in each other, as deciding Kelly-width is in  $\text{Nptime}$ . It is known that DAG-width is bounded in Kelly-width by a quadratic function [KKRS14].

Yet another way to cope with the problem shown here is to weaken the power of cops by forbidding them to be placed outside the robber component. It is easy to see that in this case the DAG decomposition resulting from a winning cop strategy (if they still can win) has size polynomial in the size of the graph. That restriction also allows one to solve the monotonicity cost problem: given a strategy for  $k$  cops that guarantees a capture of the robber, how many cops are needed to do this in a robber-monotone way? In the restricted version one can come up with a linear number of additional cops [KKRS14].

## References

- [Bar06] János Barát. Directed Path-width and Monotonicity in Digraph Searching. *Graphs and Comb.*, 22(2), 2006.
- [BDH<sup>+</sup>12] Dietmar Berwanger, Anuj Dawar, Paul Hunter, Stephan Kreutzer, and Jan Obdržálek. The DAG-width of directed graphs. *J. Comb. Theory*, 102(4):900–923, 2012.
- [Die12] Reinhard Diestel. *Graph Theory, 4th Edition*. Springer, 2012.

- [FGT11] F. Fomin, P. Golovach, and D. Thilikos. Approximation algorithms for domination search. In Klaus Jansen and Roberto Solis-Oba, editors, *Approximation and Online Algorithms (WAOA)*, volume 6534 of *Lecture Notes in Computer Science*, pages 130–141. Springer Berlin / Heidelberg, 2011.
- [FKM03] F. Fomin, D. Kratsch, and H. Müller. On the domination search number. *Discrete Applied Mathematics*, 127(3):565–580, 2003.
- [HK08] Paul Hunter and Stephan Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3), 2008.
- [JRST01] Thor Johnson, Neil Robertson, Paul Seymour, and Robin Thomas. Directed Tree-Width. *J. Comb. Theory, Ser. B*, 82(1), 2001.
- [KKRS14] Łukasz Kaiser, Stephan Kreutzer, Roman Rabinovich, and Sebastian Siebertz. Directed width measures and monotonicity of directed graph searching. arXiv, 2014.
- [KO09] S. Kreutzer and S. Ordyniak. Distance-d-domination games. In *34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, 2009.
- [Rab13] Roman Rabinovich. *Graph Complexity Measures and Monotonicity*. PhD thesis, RWTH Aachen University, 2013.
- [Ree99] Bruce Reed. Introducing directed tree-width. *Electronic Notes in Discrete Mathematics*, 3:222 – 229, 1999.
- [Saf05] Mohammad Ali Safari. D-width: A more natural measure for directed tree width. In Joanna Jedrzejowicz and Andrzej Szepietowski, editors, *MFCS*, volume 3618 of *Lecture Notes in Computer Science*, pages 745–756. Springer, 2005.